## Team Name: CodeCrafters

# Task 02 Report: Intent Classification Using LSTM

### Introduction

In this project, we developed a simple Natural Language Understanding (NLU) system for classifying intents from text input. The goal was to train a model using examples of different intents and then test it with random texts, providing intent classification along with confidence scores. Additionally, we implemented a fallback mechanism to handle cases where the confidence level does not meet a certain threshold.

### Dataset Creation

We created a dataset containing examples of different intents for a specific domain. Each intent was associated with several examples of text inputs. For example:

- Intent: Greet
  - Examples: "Hi", "How are you?", "Hello"
- Intent: Farewell
  - Examples: "Goodbye", "See you later", "Take care"
- Intent: Inquiry
  - Examples: "What's the weather like today?", "Can you tell me the time?", "Where is the nearest restaurant?"

### Model Training

We chose to use a Long Short-Term Memory (LSTM) neural network model for intent classification due to its effectiveness in capturing sequential information from text data. The training process involved the following steps:

- Tokenization: We tokenized the text inputs using the Keras Tokenizer and converted them into sequences of integers.
- Sequence Padding: We padded the sequences to ensure uniform input size for the LSTM model.
- Model Architecture: We built a Sequential model comprising an Embedding layer, a Bidirectional LSTM layer, and a Dense layer with softmax activation.
- Model Compilation: We compiled the model using sparse categorical cross-entropy loss and the Adam optimizer.
- Model Training: We trained the model on the prepared dataset for 50 epochs.

### Intent Classification Function

We developed a function to classify intents from new text inputs using the trained LSTM model. The function returns the predicted intent along with a confidence score.

### Fallback Mechanism

We implemented a fallback mechanism to handle cases where the confidence score for the predicted intent is below a predefined threshold (e.g., 0.7). If the confidence score does not meet this threshold, the system returns a fallback response indicating that the intent could not be confidently determined.

### Testing

We tested the trained model and classification function with random text inputs to ensure they provide the expected intent classifications along with confidence scores or fallback responses.

### Conclusion

In conclusion, we successfully developed a simple NLU system for intent classification using an LSTM model. The system effectively classifies intents from text inputs, providing confidence scores and fallback responses when necessary. Further improvements could include experimenting with different architectures, exploring additional preprocessing techniques, and fine-tuning the fallback mechanism for optimal performance.